NAVAL POSTGRADUATE SCHOOL

Monterey, California

THESIS

STRATEGIC ALLOCATION OF SEALIFT:
A GAMS-BASED
INTEGER PROGRAMMING APPROACH

by

Michael J. Lally

September 1987

Thesis Advisor: Richard E. Rosenthal

Approved for public release; distribution is unlimited

SECURITY CLASSIFICATION OF THIS PAGE

REPORT DOCUMENTATION PAGE

1a REPORT SECURITY CLASSIFICATION	1b RESTRICTIVE MARKINGS
2a SECURITY CLASSIFICATION AUTHORITY	3 DISTRIBUTION/AVAILABILITY OF REPORT Approved for public release; Distribution is unlimited
2b DECLASSIFICATION/DOWNGRADING SCHEDULE	
4 PERFORMING ORGANIZATION REPORT NUMBER(S)	5 MONITORING ORGANIZATION REPORT NUMBER(S)
6a NAME OF PERFORMING ORGANIZATION Naval Postgraduate School 6b OFFICE SYMBOL (If applicable) 55	7a NAME OF MONITORING ORGANIZATION Naval Postgraduate School
6c ADDRESS (City, State, and ZIP Code) Monterey, California 93943-5000	7b ADDRESS (City, State, and ZIP Code) Monterey, California 93943-5000
8a NAME OF FUNDING/SPONSORING ORGANIZATION 8b OFFICE SYMBOL (If applicable)	9 PROCUREMENT INSTRUMENT IDENTIFICATION NUMBER
8c ADDRESS (City, State, and ZIP Code)	10 SOURCE OF FUNDING NUMBERS PROGRAM ELEMENT NO / PROJECT NO / TASK NO / WORK UNIT ACCESSION NO

11 TITLE (Include Security Classification)

STRATEGIC ALLOCATION OF SEALIFT: A GAMS-BASED INTEGER PROGRAMMING APPROACH

12 PERSONAL AUTHOR(S) LALLY, Michael J.

13a TYPE OF REPORT	13b TIME COVERED	14 DATE OF REPORT (Year, Month, Day)	15 PAGE COUNT
Master's thesis	FROM ____ TO ____	1987 September	50

16 SUPPLEMENTARY NOTATION

17 COSATI CODES			18 SUBJECT TERMS (Continue on reverse if necessary and identify by block number)
FIELD	GROUP	SUB-GROUP	deployment, GAMS, Variable reduction

19 ABSTRACT (Continue on reverse if necessary and identify by block number)

This study develops a prototype model which can be used to allocate strategic sealift resources in crisis deployments. The first part of the model is a GAMS-Based Integer Program that extends a classic network flow optimization model developed by Dantzig and Fulkerson. The second part uses a Fortran program to convert the GAMS output into ship schedules. Using intelligent reduction methods, the formulation reduces the number of constraints by 60-70% and the number of variables and nonzero elements in the matrix by 90-99%. Results of this study indicate integer programming with these reduction methods is a viable alternative to modelling sealift as continuous flow variables.

20 DISTRIBUTION/AVAILABILITY OF ABSTRACT ☒ UNCLASSIFIED/UNLIMITED ☐ SAME AS RPT ☐ DTIC USERS	21 ABSTRACT SECURITY CLASSIFICATION
22a NAME OF RESPONSIBLE INDIVIDUAL Prof. Rosenthal	22b TELEPHONE (Include Area Code) 408-646- 2795 22c OFFICE SYMBOL 55Rl

DD FORM 1473, 84 MAR 83 APR edition may be used until exhausted. All other editions are obsolete

SECURITY CLASSIFICATION OF THIS PAGE

Strategic Allocation of Sealift: A GAMS-Based Integer Programming Approach

by

Michael J. Lally
Captain, United States Army
B.S., United States Military Academy, 1978

Submitted in partial fulfillment of the
requirements for the degree of

MASTER OF SCIENCE IN OPERATIONS RESEARCH

from the

NAVAL POSTGRADUATE SCHOOL
September 1987

ABSTRACT

This study develops a prototype model which can be used to allocate strategic sealift resources in crisis deployments. The first part of the model is a GAMS-Based Integer Program that extends a classic network flow optimization model developed by Dantzig and Fulkerson. The second part uses a Fortran program to convert the GAMS output into ship schedules. Using intelligent reduction methods, the formulation reduces the number of constraints by 60-70% and the number of variables and nonzero elements in the matrix by 90-99%. Results of this study indicate integer programming with these reduction methods is a viable alternative to modelling sealift as continuous flow variables.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

ACKNOWLEDGEMENTS

For their invaluable contribution towards this thesis, I would like to thank the following people:

- Dr. Alexander Meeraus and his associates at the World Bank for providing the use of a courtesy copy of the GAMS Modeling System.
- Professor Richard E. Rosenthal, my thesis advisor, for his sound advice, professional assistance and editorial comments.
- Professor Siriphong Lawphongpanich, my second reader, for his critique and final editing to improve the quality of the research paper.

I. INTRODUCTION

A. PROBLEM DESCRIPTION

The United States must be able to deploy U.S. Forces and material in support of a Commander conducting operations anywhere in the world. It is imperative that the forces and material arrive at the *time* and *location* requested. Furthermore, for a deployment to succeed the air and sealift assets must be properly allocated.

B. JOINT DEPLOYMENT AGENCY

The complexity and magnitude of deploying U.S. Forces to an overseas area requires careful and thorough coordination. In March 1979 the Joint Deployment Agency (JDA) was established to be the single point of contact for strategic deployment planning and coordination.

The Joint Deployment Agency's mission is to support the Joint Chiefs of Staff (JCS) and Supported Commanders in planning for and executing deployments. As directed by the Joint Chiefs of Staff, the JDA is responsible for coordinating the actions of deploying units and the land, air and sealift assets to be used in a joint deployment. The JDA also serves as the focal point for information associated with deployment decisions.

The land, air and sealift assets to be used in a joint deployment are controlled by three different organizations which are called the transportation operating agencies.

> The Military Sealift Command (MSC), which operates some 150 ships in support of the Navy's daily operations, has earmarked 51 additional ships for joint deployment. These are comprised of nine dry cargo ships, 22 general-purpose oil tankers, 12 maritime prepositioning ships now on station at various points around the globe and eight fast sealift ships. In wartime, this number would be supplemented on short notice by 116 ships in the Ready Reserve Fleet (RRF).
>
> The Military Airlift Command (MAC), which is responsible for all of the military's overseas air and freight transportation, will immediately contribute 234 C-141 "Starlifters" and 77 C-5 aircraft. In a protracted conflict, MAC will also have access to 238 aircraft in the Civil Reserve Air Fleet. Not included in these totals are some 500 of MAC's C-130 "Hercules" transports that are meant for supporting overseas unified commands.

The Military Traffic Management Command (MTMC) is responsible for identifying and managing the transportation routes to be used in the United States for the shipment of supplies to the ports where they are to be loaded aboard MSC's ships. MTMC will supervise the loading operations in the United States and the unloading at the destination. [Ref. 1: p. 40]

C. DEPLOYMENT PLANNING

1. Deliberate Planning

Deployment planning is normally a time-consuming and "deliberate" process. During an emergency or time-sensitive situation the JDA must be prepared to conduct deployment planning in a "crisis" mode. The deliberate planning process takes place over a period of weeks or months. The procedure insures people and material can be moved to support military objectives. Every movement plan goes through a refinement process. The most important question in the process is, "Will the deployment plan work?" Once it is determined the deployment plan can be supported with available transportation assets, the JDA looks for ways to improve the plan. Can the deployment be carried out faster, more efficiently, more cost-effectively, or with less vulnerability to enemy attack? Refinement is an ongoing process with movement requirements validated, lift assets matched against specific movement requirements and schedules updated as necessary. For each movement requirement (force, equipment or supply) an approved operations plan designates lift assets, ports of embarkation, ports of debarkation, load dates and delivery dates. If the deployment plan cannot be supported, the decision makers must make one of the following decisions. They must allocate additional lift assets, adjust required delivery dates, delete low priority movement requirements or change the operations plan.

2. Crisis Planning

The second category is crisis planning. Crisis action deployment planning requires decisions on force commitment within an initial four hour period [Ref. 2]. The Supported Commander must be able to develop a course of action which is acceptable in terms of his military objectives and is supportable by the transportation system [Ref. 3: p. 1-2]. The key question is, "Can the plan be supported?" Furthermore, guidance such as size of deploying forces, quantity of material, ports of embarkation and debarkation, required delivery dates or rates, and mode selection must be passed down to the Commands controlling land, air, and sealift assets. These Transportation Operating Agencies (MTMC, MAC, MSC) must initiate their planning processes in order to properly support the deployment in a timely manner.

For the analyst the key difference between deliberate and crisis planning is the time constraint for constructing a viable deployment plan. Any model that is used as an aid in the crisis decision making process must generate a solution in under four hours [Ref. 2].

D. INPUTS TO PROBLEM

In order to coordinate land, air and sealift assets the JDA assembles data relating to deployability of all forces, equipment and supplies.

- The first inputs are the movement requirements. A movement requirement consists of the following items:
  - Classification (passengers, equipment, fuel, etc.)
  - Weight and Quantity
  - Origin and Destination
  - Available Load Dates
  - Required Delivery Dates
- The second inputs are a listing of the ports and their capacities. Capacity is the short tons per day of material that can be moved through the airport or seaport.
- The third inputs are the lift assets.
  - Type and Quantity available
  - Capacity in short tons
  - Utilization rate
  - Transit times

Given these inputs the Joint Deployment Agency attempts to optimize strategic lift resources in support of deployments.

E. RESEARCH TOPIC

This study will develop a prototype model that can be used to allocate strategic sealift resources. The model will attempt to show that integer programming is a viable method for solving the sealift allocation problem. Furthermore, the model will incorporate intelligent reduction methods to restrict the number of equations and variables so as to reduce the size of the model. Without these reductions, the integer programming approach would not be computationally viable.

II. BACKGROUND

A. GEORGIA TECH RESEARCH GROUP

The primary research attempting to solve this large deployment problem has come from a group led by Professors John J. Jarvis and H. Donald Ratliff of the Georgia Institute of Technology. During the past five years their effort has been to:

1. examine deployment planning in a crisis action environment from a modelling perspective;
2. assess available methodology and modelling concepts for application to the crisis action environment;
3. develop concepts and methodology for closure optimization;
4. develop a system architecture within which these models would function [Ref. 3: p. 1].

Jarvis and Ratliff describe four levels of models, which if developed, would function in a hierarchy. Decisions and assumptions made at the higher levels would guide and constrain decisions at the lower levels. Violations of these constraints could not occur unless the higher level modifies or changes the constraining decision or assumptions. The lower the level the more the detail involved in the planning process.

The highest level is the closure planning level. The primary purpose of this level is to aid the decision maker in developing a general movement plan which will satisfy the military objectives and can be supported by the available transportation system. A general movement plan includes mode, port of embarkation (POE), port of debarkation (POD), assignment of movement requirements, timing of movements, degree of flexibility allowed at lower levels, and manner in which movement requirements can be split for transportation. The decisions made at this level are the most important because they guide and constrain all future decisions. [Ref. 3: pp. 7-10]

The second level is the system loading/coordination level. Its purpose is to insure efficient utilization of the transportation system in carrying out the general movement plan developed in level one. At this level they search for and attempt to resolve problem areas, and develop more detail regarding movements. Additionally, it provides information and coordination necessary for transition from the top level to the detailed scheduling by transportation operating agencies in level three. [Ref. 3: pp. 9,11]

The third level is where detailed schedules for deployment are constructed. The three transportation operating agencies, MTMC, MAC and MSC are required to construct schedules to satisfy the specifications set forth by the first and second level. These specifications include movement requirements, suggested lift asset, POE, POD, and the required delivery dates. [Ref. 3: p. 9]

Level four system is for monitoring the development and implementation of the deployment plan. This four level system is a dynamic planning system that provides for feedback, updates, and modifications as the plan proceeds. [Ref. 3: p. 10]

B. GEORGIA TECH MODEL

The main thrust of the Georgia Tech researchers has been on level one, where the general movement plan is developed. They decided the best way to solve the deployment problem was to use decomposition. They broke the problem into two subproblems, a channel configuration and a movement requirement assignment problem. The problems are connected through a set of linking constraints. The decomposition method first generates the solution to the channel configuration model. With the link constraints fixed, the movement requirement assignment problem is solved. The results of this model generate a linking constraint that is passed back to the channel configuration model which is solved again. This process is repeated until the solutions converge at optimality or it can be stopped at the user's discretion if time is limited. [Ref. 3: p. 45]

In the search for appropriate solvers Jarvis and Ratliff have attempted to take advantage of the sparseness of the network generated by a deployment problem. The movement requirement assignment problem can be formulated as a pure network structure, therefore it can be best solved using a network solver [Ref. 3: p. 49]. For the channel configuration model they chose a solver for a network with side constraints [Ref. 3: p. 245]. Due to the large number of side constraints they may switch to a linear program solver in the future [Ref. 2]. The two problems are linked together with Benders decomposition method.

C. MODEL PROBLEMS

The current model is experiencing several problems. The solution procedure adopted for the model converges slowly and at times does not produce an optimal answer. Research is ongoing in an attempt to discover the source of the convergence problem.

When the procedure produces an optimal solution, it requires over six hours even for a medium size deployment problem. This is not acceptable for crisis planning. Current research is investigating a "hot restart" capability, aggregation of movement requirements, suboptimal stopping rules, a method to generate arcs as needed, and arc reduction methods.

A third area of concern is the method of modelling sealift. The model assumes continuous flow variables even though cargos move discretely in reality. The concept can best be understood by likening the channel and its capacity to a pipe with a water passing through it at a given flow rate.

The Georgia Tech research group makes a good argument for the channel concept and continuous flow rate when applied to airlift. The airlift cycle times are relatively small when compared to the time horizon and the delivery effect is "smoothed" over time. [Ref. 3: p. 35] However, the assumption of continuous flow variables for sealift is not realistic. Travel times are not small when compared to the time-step of the model. Furthermore, the assumption of a steady-state system cannot be made during the initial 20 to 30 days and it is questionable whether the steady-state system argument can be applied to each port in the later stages of a deployment plan (30-180 days). Finally, the simplifying assumption of continuous flow variables for sealift is not necessary. This study will show that ship departures and arrivals can be modelled discretely.

D. OTHER RESEARCH

In a related Masters thesis in Operations Research, CPT. K. Steven Collier, Naval Postgraduate School, Monterey, California constructed a model for determining optimal flow rates for air and sealift. [Ref. 4]

III. MODEL DEVELOPMENT

A. INTRODUCTION

In 1954, G. B. Dantzig and D. R. Fulkerson showed that the problem of determining the minimum number of tankers required to meet a fixed schedule could be modeled and solved as a linear network flow optimization problem [Ref. 5: p. 217]. The problem assumes that the load date, origin, and destination for each shipload are known, all ships have the same capacity and speed, and all movement requirements are full shiploads. This study extends the Dantzig and Fulkerson tanker scheduling model to include time windows for loading. Instead of specifying a single load date, a time window from Earliest Load Date to Latest Load Date may be specified. The remainder of this chapter will compare the two formulations and present an example of the new one.

B. MATHEMATICAL FORMULATIONS

1. Dantzig-Fulkerson Model

Indices:

i	=	Port of embarkation (POE) index represents the port of origin for a shipload of cargo
j	=	Port of debarkation (POD) index represents the port of destination for a shipload of cargo
a	=	Loading time period for cargo
b	=	Unloading/discharge time period for cargo

Given Data:

T_{ij}	=	Loading time + transit time from POE i to POD j
TT_{ij}	=	Unloading time + transit time from POD j to POE i

Derived Data:

n_{ai}	=	Number of ships that load at POE i on day a

N_{bj} = Number of ships that unload at POD j on day b

Decision Variables:

X_{aibj} = Number of ships departing from POD j at time b to return to POE i at time a (reassignment of ship)

XX_{ai} = Number of ships starting their schedule at time a from POE i

Y_{bj} = Number of ships completing their schedule at time b at POD j

Formulation:

Maximize Z

Subject To:

$$\sum_{b,j} X_{aibj} + XX_{ai} = n_{ai} \qquad \text{for all } a,i \qquad \text{(eqn 3.1)}$$

$$\sum_{a,i} X_{aibj} + Y_{bj} = N_{bj} \qquad \text{for all } b,j \qquad \text{(eqn 3.2)}$$

$$\sum_{a,i} XX_{ai} + Z = \sum_{a,i} n_{ai} \qquad \text{(eqn 3.3)}$$

$$\sum_{b,j} Y_{bj} + Z = \sum_{b,j} N_{bj} \qquad \text{(eqn 3.4)}$$

$$\sum_{a,i,b,j} X_{aibj} = Z \qquad \text{(eqn 3.5)}$$

$$XX_{ai} \geq 0, \; Y_{bj} \geq 0, \; X_{aibj} \geq 0$$

Constraints 3.1 require every shipload of cargo be moved. The cargo may be moved by a ship on the first leg in its schedule, XX_{ai}, or a ship that is reassigned for

an additional leg in its schedule, X_{aibj}. Constraints 3.2 insure all required shiploads of cargo are delivered. The delivery may be the last stop of a ship, Y_{bj}. or one of several reassignments, X_{aibj} within its schedule. Constraints 3.3 and 3.4 insure that there are sufficient number of ships to satisfy all pickup and delivery requirements. The objective function maximizes the total ships reassigned, which is equivalent to minimizing the number of new ships used.

For an X_{aibj} to be feasible the arrival day a at POE i must be greater then or equal to the unloading day b at POD j added to the transit time between the two ports. It follows that if

$$a < b + TT_{ij}$$

then X_{aibj} is infeasible and X_{aibj} should be fixed to zero or, preferable deleted from the problem. This condition is not an explicit constraint in the model, but rather the relationship is used to reduce the set of X_{aibj}'s to insure only feasible values are defined in the model.

The Dantzig and Fulkerson formulation is totally unimodular, therefore the optimal solution is an integral solution.

2. Extended Formulation with Time Windows

As stated earlier, the new formulation extends the Dantzig and Fulkerson tanker scheduling model to include time windows for loading. Instead of specifying a single load date, a time window from Earliest Load Date to Latest Load Date may be specified. This situation can be equivalently modelled using time windows for unloading or arrival dates. The time window would specify Earliest Arrival Date to Latest Arrival Date. With the addition of the time windows, the extended formulation is no longer totally unimodular, therefore you are not guaranteed an integral solution. To insure an integral solution it is necessary to use an integer program solver. The integer program solver enables the user to model ship flows discretely versus the Georgia Tech model which assumes continuous flow variables. Similar to the original formulation, there is a specified origin and destination for each shipload and all ships are considered equal in capacity and speed.

Using one type ship in the model versus choosing between several different types of ship is acceptable at the strategic level. At this level the objective is to determine if a deployment plan is feasible. If the plan is feasible, a lower level will

implement the actual movement plan using more detail. At the strategic level cargo is broken into three categories; unit equipment, resupply and ammunition. For each category a specific ship type is designated. Unit equipment will make maximum use of roll-on roll-off ships. Resupply cargo and ammunition will use containerships and breakbulk cargo ships. The mix between the latter two is determined by the theatre of operations and the extent destination ports can handle containerships. The model can be run for each type ship and category of cargo. Any movement requirements not moved by the allocated lift assets in the first run, can be included with cargo in the next category during subsequent runs.

Indices:

i	=	Port of embarkation (POE) index represents the port of origin for a shipload of cargo
j	=	Port of debarkation (POD) index represents the port of destination for a shipload of cargo
a	=	Loading time period for cargo
b	=	Unloading/discharge time period for cargo

Given Data:

T_{ij}	=	Loading time + transit time from POE i to POD j
TT_{ij}	=	Unloading time + transit time from POD j to POE i
REQ_{siaj}	=	1, if shipload s is required to load at POE i on day a with destination POD j 0, otherwise

Derived Data:

POE_{si}	=	1, if shipload s departs POE i 0, otherwise
POD_{sj}	=	1, if shipload s arrives POD j 0, otherwise
O_{sai}	=	1, if shipload s loads at POE i on day a 0, otherwise

D_{sbj} = 1, if shipload s unloads at POD j on day b
0, otherwise

$LINK_{sij}$ = 1, if shipload s goes from POE i to POD j
0, otherwise

Decision Variables:

X_{aibj} = Number of ships departing from POD j at time b to return to POE i at time a (reassignment of ship)

XX_{ai} = Number of ships starting their schedule at time a from POE i

Y_{bj} = Number of ships completing their schedule at time b at POD j

M_{sai} = 1, if shipload s departs POE i on day a
0, otherwise

Formulation:

Maximize Z

Subject To:

$$\sum_{b,j} X_{aibj} + XX_{ai} = \sum_{s} M_{sai} \qquad \text{for all a,i} \qquad \text{(eqn 3.6)}$$

$$\sum_{a,i} X_{aibj} + Y_{bj} = \sum_{s,i,a} M_{sai} \qquad \text{for all b,j} \qquad \text{(eqn 3.7)}$$

$$\sum_{a,i} M_{sai} = 1 \qquad \text{for all s} \qquad \text{(eqn 3.8)}$$

$$\sum_{a,i} XX_{ai} + Z = \sum_{s,i} POE_{si} \qquad \text{(eqn 3.9)}$$

$$\sum_{b,j} Y_{bj} + Z = \sum_{s,j} POD_{sj} \qquad \text{(eqn 3.10)}$$

$$\sum_{a,i,b,j} X_{aibj} = Z \qquad \text{(eqn 3.11)}$$

$$XX_{ai} \geq 0, \; Y_{bj} \geq 0, \; X_{aibj} \geq 0$$
$$M_{sai} = 0 \text{ or } 1$$

Constraints 3.6 require every shipload of cargo be moved. The cargo may be moved by a ship on the first leg in its schedule, XX_{ai}, or a ship that is reassigned for an additional leg in its schedule, X_{aibj}. Constraints 3.7 insure all required shiploads of cargo are delivered. The delivery may be the last stop of a ship, Y_{bj}, or one of several reassignments, X_{aibj} within its schedule. Constraints 3.8 are generalized upper bounds that pick one load date from the time window of available load dates. Constraints 3.9 and 3.10 insure that there are sufficient number of ships to satisfy all pickup and delivery requirements. The objective function maximizes the total ships reassigned. the movement requirements.

Oftentimes, in a crisis deployment, the objective function is to minimize the time required to complete all the movement requirements with the available resource. In this model, the movement requirements have specific times they must depart in order to arrive at the destination and satisfy the Commander's military objectives. Therfore, the optimal solution of this model determines the minimum number of ships required to satisfy all the movement requirements. Given sufficient assets the model will determine if there is a way to satisfy the requirements with fewer assets. If there are not sufficient sealift assets available, the Commander must allocate additional lift assets, adjust required delivery dates or change the operations plan.

As in the Dantzig and Fulkerson formulation, for an X_{aibj} to be feasible,

$$a \geq b + TT_{ij}.$$

Figures 3.1 and 3.2 are network diagrams which depict the variables used in the formulation. The general network diagram with all possible arcs is shown in Figure 3.1. Figure 3.2 illustrates the effect of variable reduction methods that will be discussed in Chapter IV. The problem that generates these network diagrams is discussed in Section C.

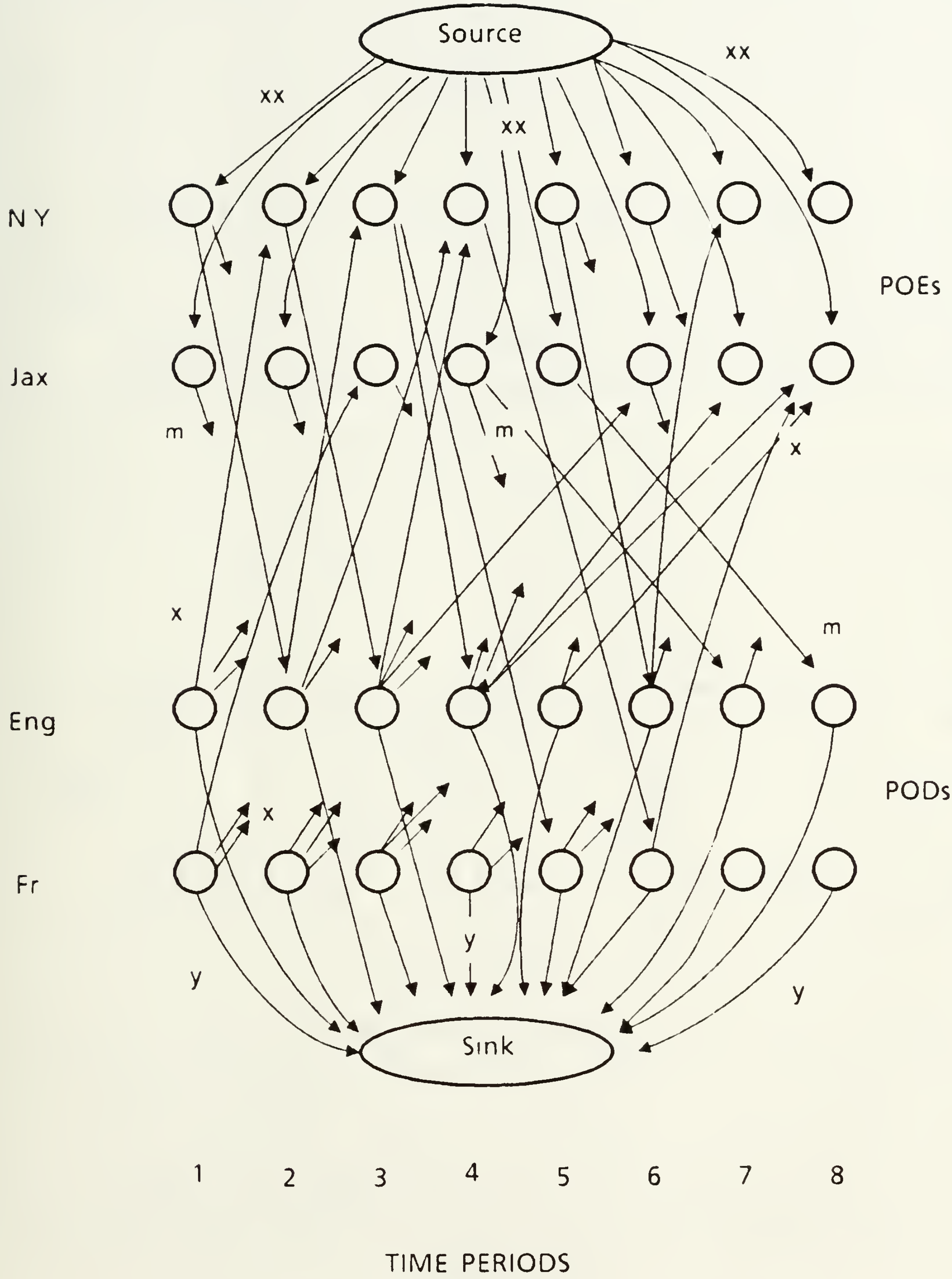

Figure 3.1 General network diagram with all possible arcs and nodes.

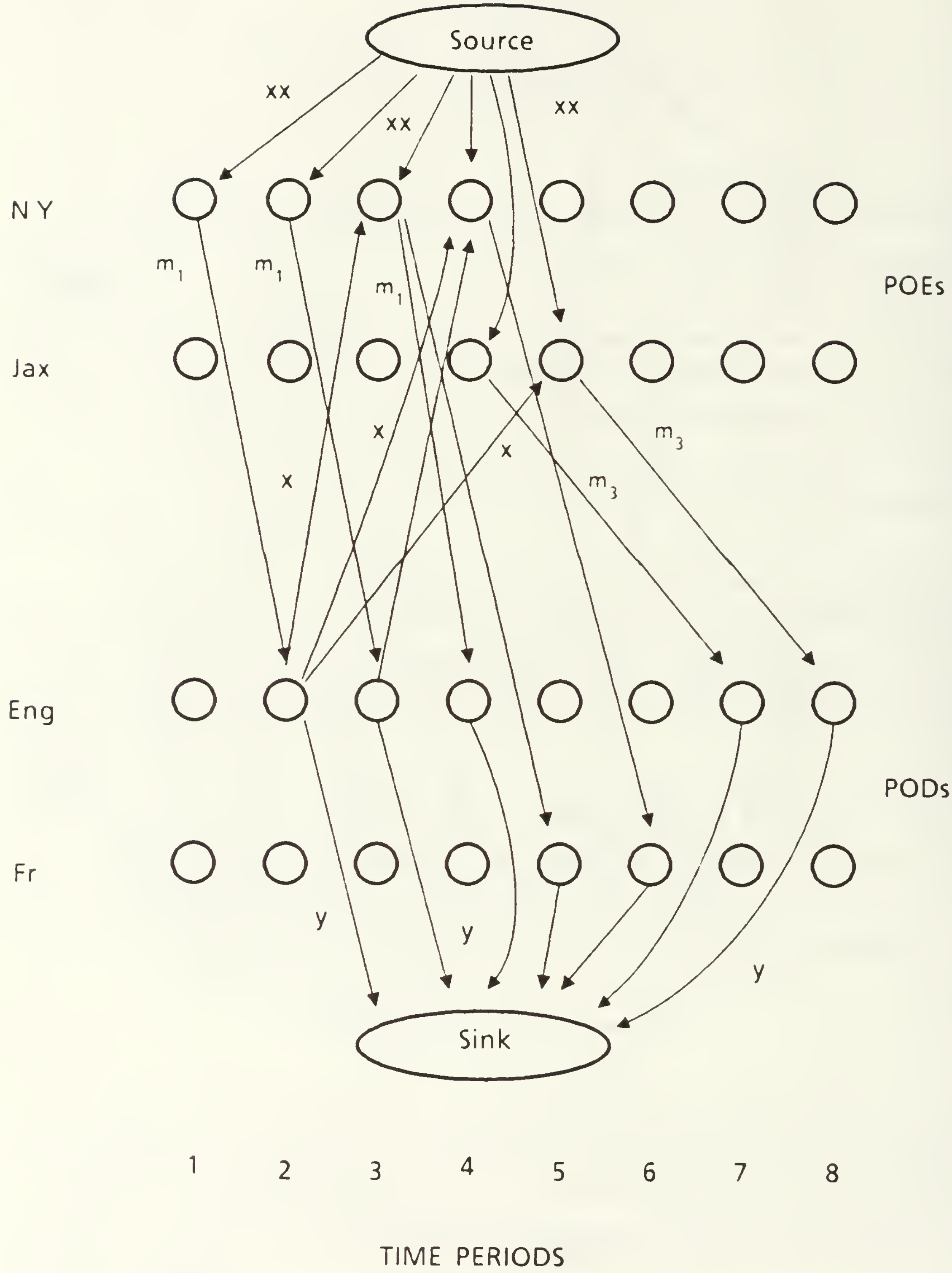

Figure 3.2 Network diagram with reduction methods incorporated.

The new ships, XX_{ai}, are arcs with tail at the source and head at node a,i. The ships that stop, Y_{bj}, are arcs with tail at node b,j and head at the sink. The reassigned ships, X_{aibj}, are arcs with tail at node b,j and head at node a,i. The binary variable M_{sai} can be represented as an arc with tail at origin node a,i and head connected to the correct destination, node b,j, where $D_{sbj} = 1$.

The key difference between this formulation and the original Dantzig and Fulkerson formulation is the introduction of the binary variable M_{sai}. For example, if shipload 1 is available to load at New York on day 1, 2 or 3 there will be an M_{sai} for s = 1, a = 1, 2, and 3, and i = NY (See Figure 3.2). Constraints 3.8 select the one M_{sai} that represents the best load date from the time window of available dates. Constraints 3.6 insure that for every arc M_{sai} leaving node a,i there is an arriving arc X_{aibj} or arc XX_{ai} as shown in Figure 3.3 .

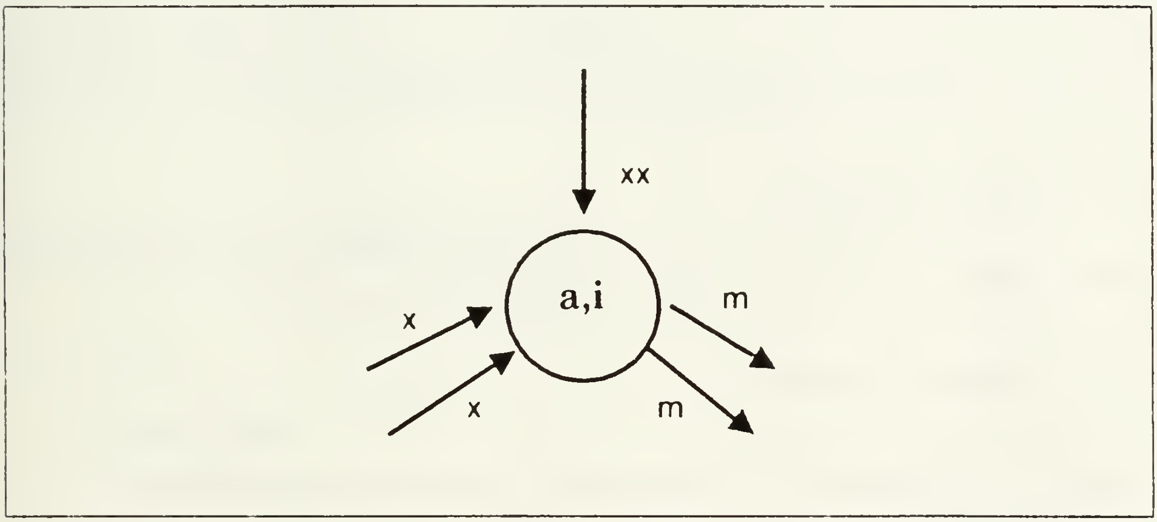

Figure 3.3 Constraints 3.6.

Constraints 3.7 insure that for every arc M_{sai} arriving at node b,j there is a departing arc X_{aibj} or arc Y_{bj} as shown in Figure 3.4 .

The binary variable M_{sai} is not directly linked to a node b,j, therefore it is necessary to find a method to link M_{sai} to the correct node, where $D_{sbj} = 1$. The correct node b,j is determined by carefully defining a subset for each index s,i and a. The righthand side of Constraints 3.7 is now rewritten more precisely as follows;

$$\sum_{s \in D_{sbj}} \sum_{i \in POE_{si}} \sum_{a \in a'} M_{sai} \qquad \text{for all b,j.}$$

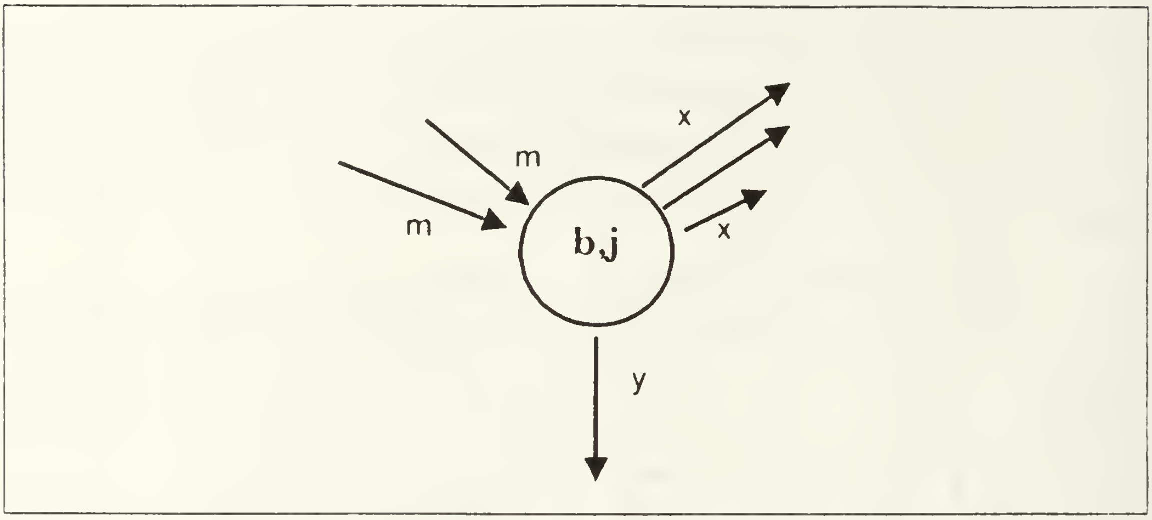

Figure 3.4 Constraints 3.7.

The above notation, using GAMS inspired shorthand, reads, "Summing over all s such that $D_{sbj} > 0$; Summing over all i such that $POE_{si} > 0$; Summing over a such that a = b - T_{ij}." Examples of D_{sbj} and POE_{si} are found in Section C2.

C. EXAMPLE PROBLEM

To illustrate the model, consider the following numerical example for which the corresponding network diagram is depicted in Figure 3.2. Let there be

- 2 POEs, i = NY, Jax
- 2 PODs, j = Eng, Fr
- 3 shiploads
- 10 time periods.

1. Given Data

There are three tables of data required to run the model. In the example we assume the transit times are equal in both directions, $T_{ij} = TT_{ij}$. Therefore, the model input will consist of two sets of data. Table 1 lists the time it takes a ship to load or unload its cargo and travel from port i to port j. The travel times are purposely kept small.

The Movement Requirements Table (REQ) lists each shipload, port of embarkation, available to load dates, and port of debarkation. Following the notation used in the GAMS modeling language [Ref. 6,7,8,9,10] Table 2 reads, "Shipload 1 is available for loading in New York on days 1 through 3 with a destination of England."

TABLE 1

T_{ij} TIMES (DAYS)

	Eng	Fr
NY	1	2
Jax	3	2

Each shipload must have one POE i and one POD j . Our GAMS implementation of the model contains a check for this condition. If an error is detected the program will stop immediately and print several error messages to help locate the mistake.

TABLE 2

MOVEMENT REQUIREMENTS

	Eng	Fr
S01.NY.T01*T03	1	
S02.NY.T03*T04		1
S03.Jax.T04*T05	1	

2. Derived Data

After checking for errors in the REQ Table several other tables of information are derived from the REQ Table. The information is derived through a process that Kendrick and Meeraus call projection. The GAMS modeling language stores data as a relational database. "Projections can be used to answer queries by projecting the n dimensional data in a relationship onto an n-1 dimensional or smaller space." [Ref. 8: ch. 19, p. 3] The derived data is used to reduce the size of the model and is used in the procedure that determines the schedule for each ship. Listed in the following tables are examples of the derived data.

Table 3 is an example of projecting the four dimensional data in the REQ_{siaj} Table onto the three dimensional space of the O_{sai} Table. Table 3 reads, "Shipload #1 is available for loading in New York on days 1, 2 or 3. Shipload #2 is available for loading in New York on days 3 or 4."

TABLE 3

Example of O_{sai}

	NY	Jax
S01.T01	1	
S01.T02	1	
S01.T03	1	
S02.T03	1	
S02.T04	1	
S03.T04		1
S03.T05		1

Table 4 gives possible dates for unloading. For example, the first five rows read, "Shipload #1 unloads in England on day 2, 3 or 4. Shipload #2 unloads in France on day 5 or 6."

TABLE 4

Example of D_{sbj}

	Eng	Fr
S01.T02	1	
S01.T03	1	
S01.T04	1	
S02.T05		1
S02.T06		1
S03.T07	1	
S03.T08	1	

Table 5 determines the connection between the origin and destination. It reads, "Shipload #1 departs New York with a destination of England. Shipload #2 departs New York with a destination of France."

TABLE 5

Example of $LINK_{sij}$

	Eng	Fr
S01.NY	1	
S02.NY		1
S03.Jax	1	

Table 6 reads, "Shipload #1 and #2 load in New York and Shipload #3 loads in Jacksonville."

TABLE 6

Example of POE_{si}

	NY	Jax
S01	1	
S02	1	
S03		1

Table 7 reads, "Shipload #1 and #3 unload in England and Shipload #2 unloads in France."

TABLE 7

Example of POD_{sj}

	Eng	Fr
S01	1	
S02		1
S03	1	

IV. MODEL REDUCTION

As formulated above, the problem includes a large number of variables and constraints even for a medium size problem. This, in practice, means that to obtain a solution would require a large amount of computer memory and cpu time. Additionally, for a large problem, the model may be too large for most computers. Therefore, it is important to develop intelligent methods to reduce the number of variables and equations used in the model. This chapter will discuss several reduction methods that make the new model computationally viable.

The General Algebraic Modelling System (GAMS), developed by Alexander Meeraus and Tony Brooke, provides modelers with a tool to eliminate all variables unnecessary for the problem in an efficient manner, which reduces the size of the resulting optimization problem. An important feature is the $ operator, which translates to "such that". The $ operator can be used to restrict the domain of variables, reduce the number of equations generated, or specify that the summation should be only over a defined subset of indices. The remainder of this section will use GAMS notation to explain model reduction methods.

Constraint 3.8 provides a excellent example to illustrate the use of GAMS notation. Constraint 3.8 is referred to as LOAD(s) in the GAMS formulation. LOAD is the "name" of the equation and the set of shiploads s is the "domain" of this equation. If Constraint 3.8 were written for all s it would appear as follows;

```
LOAD(s) ... SUM((a,i), M(s,a,i)) =E= 1.
```

This reads, "Generate a LOAD equation for each s. In each equation the sum over a,i for all M_{sai} equals one." In the formulation it is important to accurately define the feasible set for the binary variable M_{sai}. The smaller the set of binary variables, the faster the problem can be solved. Feasible M_{sai}'s are possible only when there is a shipload s available for loading at POE i on day a. Therefore, the set for defined variables M_{sai} is restricted by using the data derived for the O_{sai} Table. Constraint 3.8 is now written;

```
LOAD(s) ... SUM((a,i), M(s,a,i) $ O(s,a,i)) =E= 1.
```

This reads, "...Sum over a,i all M_{sai} such that s,a,i belongs to the set defined by O(s,a,i) > 0."

A. VARIABLE REDUCTION

In 1954 the Dantzig-Fulkerson formulation was designed to be solved manually using the simplex algorithm. They identified the feasible subset of X_{aibj} by creating a tableau and crossing out all cells containing infeasible X_{aibj}. With GAMS notation this can be accomplished with the following lines;

```
SET  FEASIBLE                    feasible reassignments for X;
          FEASIBLE(a,i,b,j) = YES $ (ord(a) - ord(b) GE T(i,j)).
```

This reads, "A subset (called FEASIBLE) of the indices a,i,b,j is defined for only those combinations of a,i,b,j such that the ordinal position of a minus the ordinal position of b is greater than or equal to T_{ij} ($a - b \geq T_{ij}$)." With this set defined it is now possible to generate equations that include only the feasible variables. For example, Constraint 3.11 becomes

```
SUM((a,i,b,j), X(a,i,b,j) $ FEASIBLE(a,i,b,j)) = Z.
```

This reads, "Sum over a,i,b,j all X_{aibj} such that a,i,b,j is in the subset FEASIBLE ... "

A second method to reduce the set of X_{aibj} is

```
SET PORT
        PORT(a,i,b,j) = YES $ SUM(s, O(s,a,i)) $ SUM(s, D(s,b,j)).
```

Each X_{aibj} is an arc with tail at node b,j and head at node a,i. Set PORT forces every arc X_{aibj} to have a tail where a ship has delivered cargo. If no ship had delivered cargo to that port there would be no ship available for a return trip. Furthermore, Set PORT forces every arc X_{aibj} to have a head where there is a shipload ready to be loaded. There is no reason to send a ship to a POE if it is not needed at that location.

A third method to reduce the set X_{aibj} is based on common sense and experience, and is controlled by the user of the model.

```
SET   RESTRICT
   RESTRICT(a,i,b,j) = YES $ (ord(a) - ord(b) LE (T(i,j) + SLACK))).
```

The value for variable SLACK is set by the user of the model. A ship must be reassigned to a POE i within a specified length of time (transit time + slack time). If the ship is not reassigned within the time period it will stop. This is indicated by a value for Y_{bj}. A ship will always be reassigned if possible because the objective function maximizes reassignments. A requirement outside the time period will use a new ship. When looking at ship schedules it will be obvious if two schedules could be satisfied by one ship. The final time period in one schedule will be less than the initial time period in another schedule. The use of set Restrict is acceptable because one can assume a ship returned to its homeport if not reassigned in a timely manner. In the future the ship would be available at a new location when needed.

In the formulation all three set reduction methods are combined into

```
SET   OK                                    X values allowed in the model
      OK(a,i,b,j) = YES S FEASIBLE(a,i,b,j)
                             S PORT(a,i,b,j)
                                      S RESTRICT(a,i,b,j).
```

Constraint 3.11 becomes

```
SUM((a,i,b,j), X(a,i,b,j) S OK(a,i,b,j)) = Z.
```

B. EQUATION REDUCTION

In addition to reducing variables it is important to reduce the number of equations in a large model. Constraint 3.6 is specified above for every possible combination of a and i. However, it is only necessary to generate equations when there is the possibility of some shipload s loading at POE i on day a. With GAMS, we can reduce the number of equations by generating Constraint 3.11 (or POECAP(a,i) as it is called in the GAMS formulation) only when needed. We do this by applying a S operator to the equation definition;

```
POECAP(a,i) S SUM(s, O(s,a,i)) ...
```

This is interpreted , "Generate an equation for each (a,i) pair such that the sum over s of O_{sai} is non-zero." This condition will be true and hence the equation will be generated only when at least one shipload is eligible to leave POE i on day a. The remainder of the equation can be seen in Appendix A.

Similarly, Constraint 3.7 should only be generated when there is the possibility of a shipload s unloading at POD j on day b. Therefore Constraint 3.7 , which is referred to as PODCAP(b,j) in the GAMS formulation, is restricted as follows;

```
PODCAP(b,j) $ SUM(s, D(s,b,j)) ...
```

The interpretation is similar to Constraint 3.6 .

C. GAMS SOLUTION TO THE EXAMPLE PROBLEM

The optimal solution of this formulation consists of four sets of variables. Also listed is the solution of the example problem. The network diagram for this optimal solution is shown in Figure 4.1.

- XX_{ai}, a schedule is constructed for each new ship.

```
XX(T01.NY) = 1
XX(T04.Jax) = 1
```

- X_{aibj} is a reassignment of a ship from POD j day b to POE i day a.

```
X(T03.NY.T02.Eng) = 1
```

- Y_{bj}, a schedule is completed for the ship at POD j on day b.

```
Y(T05.Fr) = 1
Y(T07.Eng) = 1
```

- M_{sai} is the connection between POE i and POD j.

```
M(S01.T01.NY) = 1
M(S02.T03.NY) = 1
M(S03.T04.Jax) = 1
```

The output is printed to a file which is then input to a Fortran program, which constructs a schedule for each ship.

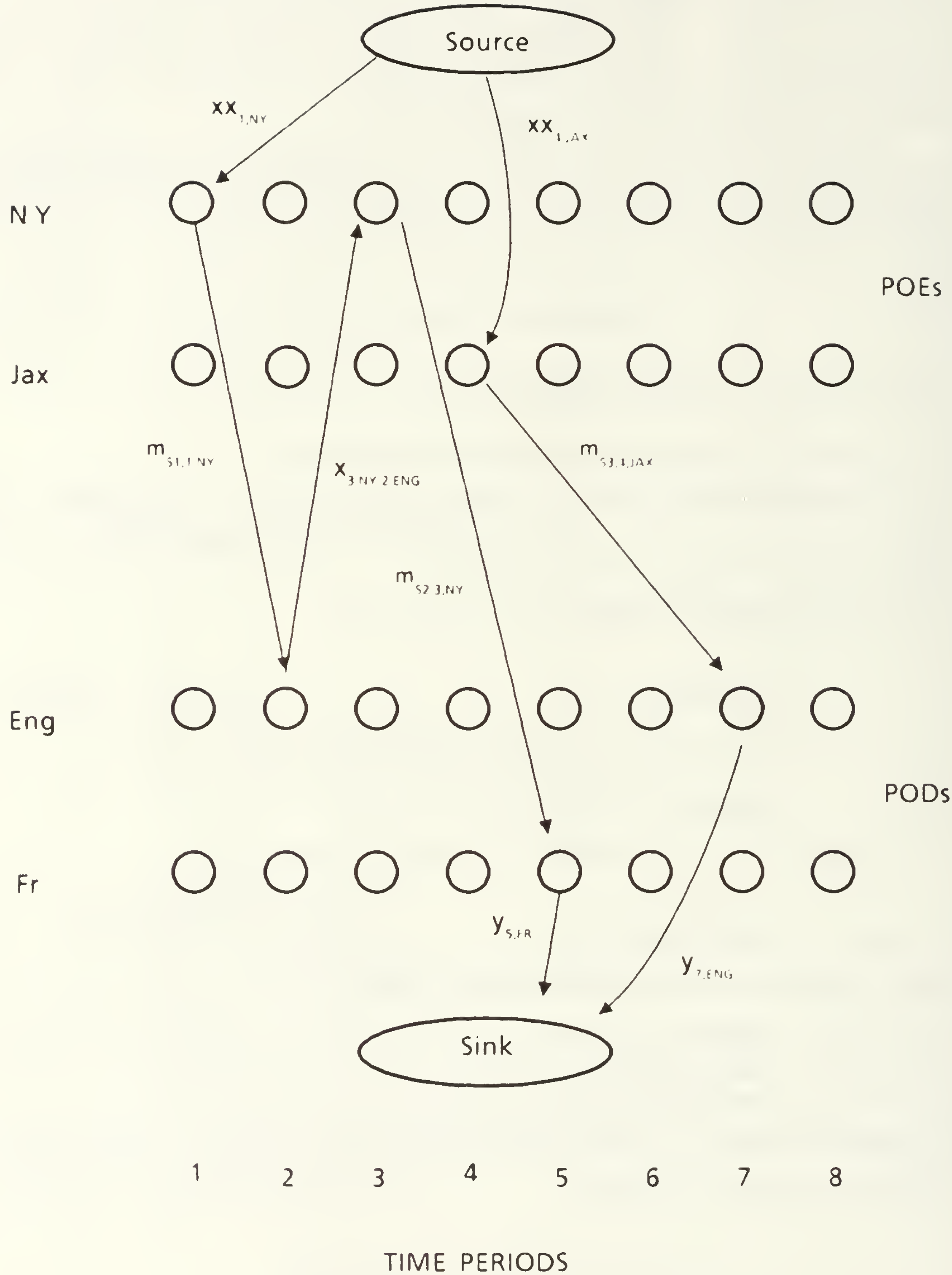

Figure 4.1 Optimal solution to the example problem.

V. SCHEDULE PROCEDURE

Dantzig and Fulkerson developed a procedure to turn their solution into a schedule for each ship [Ref. 5: pp. 219-220]. We present a new procedure, implemented in a FORTRAN program called SCHED, that is an adaptation to accommodate the binary variables M_{sai}. Given integral solutions to the four sets of variables from the GAMS model, SCHED will construct a schedule for each ship. SCHED has two major components. Part 1 reads in the data from an output file generated by the GAMS model. If the solution is non optimal, infeasible, or unbounded the program will stop and print an error message. If the solution is optimal it goes to Part 2 which contains the following procedure for determining a schedule for each ship.

Ship Schedule Procedure

Input: Integer Values for x_{aibj}, xx_{ai}, y_{bj}, and m_{sai}

Output: Ship Schedules

For all $xx_{ai} \geq 0$

{ Select any a′,i′

    Print ("Depart", i′, "Day", a′)

    Select any s′ such that $m_{s'a'i'} = 1$

        $xx_{a'i'} = xx_{a'i'} - 1$

        $m_{s'a'i'} = m_{s'a'i'} - 1$

        Select any j′ such that $LINK_{s'i'j'} = 1$

            $b' = T_{i'j'} + a'$

            Print ("Arrive", j′, "Day", b′)

            Do

                { $X = \{ x_{aib'j'}$ for all a,i: $x_{aib'j'} \geq 0 \}$

                While ($X \neq \emptyset$)

                    Select any a′,i′ such that $x_{a'i'b'j'} \in X$

                        Print ("Depart", j′, "Day", b′/"Arrive", i′, "Day", a′)

                        Select any s′ such that $m_{s'a'i'} = 1$

                            $x_{a'i'b'j'} = x_{a'i'b'j'} - 1$

                            $m_{s'a'i'} = m_{s'a'i'} - 1$

                            Select any j′ such that $LINK_{s'i'j'} = 1$

```
                    b′ = T_{i′j′} + a′
                    Print ("Arrive", j′, "Day", b′)
        }
        Print ("stop")
        y_{b′j′} = y_{b′j′} - 1
}
```

It is possible to generate several different schedules from a GAMS solution. However, the total number of ships utilized is not changed. The automation of the process, of converting GAMS output to SCHED input, makes this ship schedule procedure very easy to use.

A. SCHEDULE OUTPUT

The output from SCHED is a listing of the schedules for each ship. The output from the example problem is shown in Table 8.

TABLE 8
OPTIMAL SHIP SCHEDULES FOR EXAMPLE PROBLEM

Ship #1		PORT	PERIOD	SHIPLOAD
	Depart	NY	1	1
	Arrive	Eng	2	
	Depart	Eng	2	
	Arrive	NY	3	
	Depart	NY	3	2
	Arrive	Fr	5	
				stop
Ship #2		PORT	PERIOD	SHIPLOAD
	Depart	Jax	4	3
	Arrive	Eng	7	
				stop.

B. TIME HORIZON

A method that could be used to reduce the size of models with long time horizons is to increase the size of the time period. This reformulation would require the interpretation of the data, indices and variables to all change. Changing the time unit from 1 day to 2 or 3 days for large sealift model would be reasonable. However, a time unit of four days or more is difficult to justify. The longer time periods give less resolution to the model. Furthermore, the solutions are more general and harder to interpret.

VI. RESULTS

A. IMPLEMENTATION

GAMS was initially used to develop the model. During the development stage it saved time because it enables the user to input models using algebraic notation. This permits one to make changes and experiment with different concepts without becoming bogged down in programming a model generator. The error messages generated from mistakes or errors in logic were helpful in debugging and correcting problems. Additionally, the use of the S operator, translated to "such that", enabled the model to restrict the domain of variables, reduce unnecessary equations, specify subsets for indices and perform data transformations.

Once the problem was formulated it was necessary to choose an integer solver. GAMS is able to interface with the integer solver ZOOM, therefore GAMS was also used to generate the model.

ZOOM/XMP is a mainframe and PC computer code for solving zero/one mixed-integer programming problems developed by Roy E. Marsten at the University of Arizona. [Ref. 11] ZOOM solves the model generated by GAMS and its output is written to a formated file which serves as input to the Fortran program, SCHED.

SCHED, the Fortran program that constructs schedules for each ship is written in FORTRAN 77 and compiled by the IBM VS FORTRAN complier. It takes input from GAMS/ZOOM and constructs a listing of schedules.

B. COMPUTATIONAL RESULTS

Modeling the allocation of strategic sealift using integer programming in conjunction with intelligent reduction methods is viable. Integer programming enables the user to model discrete shipments versus modelling a continuous "flow" of cargo into sea ports. Problems involving 120 binary variables have been solved with GAMS/ZOOM in 3.6 minutes on an IBM AT with math coprocessor running at 8 megahertz.

Intelligent reduction methods are critical to solving large problems. Restricting the domain of variables and deleting unnecessary nodes and arcs in the model reduced the size of the problem. The following two examples illustrate the impact of the reduction methods.

Example Problem (Data given in Chapter 3C):

- 2 POEs
- 2 PODs
- 3 shiploads
- 10 time periods
- SLACK = 5 days

	Before Reduction	After Reduction	Percent Reduction
Constraints	46	19	.59
Continuous Variables	440	18	.96
Binary Variables	60	7	.88
Non-zero Elements	1260	62	.95

Example 2 (Data given in Appendix B):

- 5 POEs
- 5 PODs
- 30 shiploads
- 60 time periods
- SLACK = 5 days

	Before Reduction	After Reduction	Percent Reduction
Constraints	633	203	.68
Continuous Variables	90,600	589	.99
Binary Variables	9000	96	.99
Non-zero Elements	298,000	1885	.99

This significant reduction in the number of variables, equations and nonzeroes makes the difference between success and failure of the integer programming formulation. As the time horizon is extended the user controlled restriction of X_{aibj}'s, via the SLACK parameter has an even greater impact. The SLACK parameter requires that a ship be reassigned within a specified length of time. If it is not reassigned within the time interval it will stop. Too small a value may over restrict the

problem and result in poor solutions. Too large a value will not effect the solutions, but will fail to reduce the size of the problem. Experience and careful consideration must be used in choosing an appropriate value for SLACK.

VII. CONCLUSIONS

The transportation community has expended a tremendous amount of time and effort attempting to efficiently allocate lift assets. In the past many of the efforts have been disorganized and geared to specific problems. The Unified Transportation Command, which will soon replace the Joint Deployment Agency, has a goal to insure that the data bases and computer models will be coordinated at every level [Ref. 1: p. 44]. Under the guidance of the Transportation Command the research effort in large deployments will continue.

Results of this study indicate that integer programming solvers used in conjunction with intelligent model reduction methods can find solutions to deployment problems. These solutions can be used to aid in the planning and execution of deployments. Furthermore, integer programming with the resulting discrete ship flows is a viable alternative to the Georgia Tech method of modeling sealift as continuous flow variables.

There are several areas for future research that can improve the model.

- Generate sensitivity analysis information on the time windows. The insights gained would be invaluable both in refining existing deployment plans and developing new deployment plans.
- Develop the capability to generate alternate optimal solutions. Alternate solutions would allow the deployment planning process to be more flexible.
- Develop a model that allows a mix of ships with different capacities and speeds. The model would then determine the best ship for a specific movement requirement.
- Permit multiple loading (POE) and discharge ports (POD) for a single shipload. A ship could pick up cargo at two or more locations and deliver to several discharge ports.
- Rather then aggregate movement requirements into shiploads, it might be better to match individual movement requirements against an asset. This idea enables the model to consider cargo compatibility and realistic groupings of movement requirements.

A potentially promising approach that can be used to implement many of these ideas is to use column generation [Ref. 12]. A ship schedule generator could portray realistic shiploads and include any restrictions on ship type, capacity, speed, port access or cargo compatibility. In addition to answering the strategic allocation problem, a

column generation formulation would benefit the Military Sealift Command (MSC) in the scheduling of individual ships. The increased realism and the numerous parameters and constraints that can be modelled with an intelligent generator would increase user "faith" in the solution.

APPENDIX A
GAMS FORMULATION

```
$title EXAMPLE PROBLEM
$offupper
$offsymlist offuellist offuelxref offsymxref
$ontext

   Reference:
   1.  "Minimizing the Number of Tankers to Meet a Fixed Schedule",
   G.b. Dantzig and D.R. Fulkerson, Naval Research Logistics
   Quarterly, vol. 1, #3, pp.217-222, September, 1954.

   Comment:  This is a prototype model which can be used to allocate
   strategic sealift resources. It is a GAMS-Based Integer Program that
   extends a classic network flow optimization model developed by Dantzig
   and Fulkerson.  Using intelligent reduction methods, the formulation
   reduces the number of constraints by 60-70% and the number of
   variables and nonzero elements in the matrix by 90-99%.

$offtext

SETS i   POE  /NY,Jax/
     j   POD  /Eng,Fr/
     s   SHIPLOADS/S01*S03/
     TI  TIME INTERVAL/T01*T10/

ALIAS (a,b,TI);

TABLE  T(i,j)      TRAVEL TIME BETWEEN PORTS

                          ENG     FR
                NY         1       2
                Jax        3       2

TABLE   REQ(s,i,a,j)        REQUIREMENTS TABLE

                               ENG     FR
            S01.NY.T01*T03       1
            S02.NY.T03*T04              1
            S03.Jax.T04*T05      1

$ontext

     The parameter and scalar values below are used to insure no
     mistakes were made when inputting the data into the REQ Table.
     if a mistake is detected an error message will be written.

     a) Check(s) = 1, implies 1 destination was given  (GOOD).
     b) Check(s) = 2, implies 2 destinations were given  (ERROR).
     c) if check(s) for a shipload is not listed that implies no
        destination was given (ERROR).

$offtext

PARAMETER  CHECK(s)    insure each shipload has a unique destination;
      CHECK(s) =  SUM(j $ (SUM((i,a),REQ(s,i,a,j))),1);

SCALAR  ERROR       shiploads do not have a unique destination;
      ERROR = 1 $ (SUM(s $ (CHECK(s) NE 1),1));

DISPLAY $ (error) "One or more shiploads does not have a unique POD.";

DISPLAY $ (error)  CHECK;

ABORT $ (error) "Find the error and correct the REQ Table.";

$ontext

    The below parameters are generated from the above REQ Table.
    They are used to restrict the size of the model and are used
```

```
   in the algorithm that determines the schedules.
$offtext
PARAMETER   O(s,a,i)       shipload a loads at i at time a;
      O(s,a,i) = 1 $ SUM(j,REQ(s,i,a,j));
PARAMETER   D(s,b,j)       shipload s unloads at j at time b;
      LOOP(b,D(s,b,j) =
             1 $ SUM(i,REQ(s,i,b - T(i,j),j)));
PARAMETER   LINK(s,i,j)    shipload s goes from i to j;
      LINK(s,i,j) = 1 $ SUM(a,REQ(s,i,a,j));
PARAMETER  POE(s,i)       shipload s loads at POE i;
      POE (s,i) = 1 $ SUM(a,O(s,a,i));
PARAMETER  POD(s,j)       shipload s unloads at POD j;
      POD (s,j) = 1 $ SUM(b,D(s,b,j));
$ontext
     The below sets reduce the size of the problem by deleting
     infeasible and unwanted variables.
$offtext
SET FEASIBLE          feasible reassignments;
          FEASIBLE(a,i,b,j) =  YES $ (ord(a) - ord(b) GE T(i,j));
SET  PORT             include only necessary ports;
          PORT(a,i,b,j) = YES $ SUM(s,O(s,a,i))
                                             $ SUM(s,D(s,b,j));
SET REALISTIC         connections for specific trip;
          REALISTIC(a,i,b,j) = YES $ (ord(b) - ord(a) EQ T(i,j));
$ontext
   The value for SLACK is set by the user of this model.  A ship
   will be reassigned to a port within a specified length of
   time(travel time + slack time).  if the ship cannot be
   reassigned within the specified time period the ship will stop.
   a requirement outside the time period will use a new ship.
   When determining ship schedules it will be obvious if two
   schedules could be combined into one schedule.  Y(b,j) in
   a schedule will be less than XX(a,i) in another schedule.
$offtext
SCALAR SLACK  slack time for ship /5/;
SET RESTRICT          user can reduce the number of reassignments;
      RESTRICT(a,i,b,j) =  YES $ (ord(a) - ord(b) LE T(i,j) + SLACK);
SET   OK       X values allowed in model;
         OK(a,i,b,j) = YES $ FEASIBLE(a,i,b,j)
                                     $ RESTRICT(a,i,b,j)
                                        $ PORT(a,i,b,j);
VARIABLES
       X(a,i,b,j)   # Reassignments from j at time b to i at time a
       XX(a,i)      # Ships starting at time a from i
       Y(b,j)       # Ships stopping at time b at j
       Z            total return trips
       M(s,a,i)     binary variable to choose best load date
       POSiTIVE VaRIAbLES X, XX, Y;
       BINARY VARIABLE M;
*M.up(s,a,i) = 1;
*M.lo(s,a,i) = 0;
EQUATIONS
          POECAP(a,i)        POE supply
          PODCAP(b,j)        POD demand
          START              total ships starting
          FINISH             total ships finishing
          LOAD(s)            GUB for loading
          ANS                Objective function;
POECAP(a,i) $ (SUM(s,O(s,a,i)))..
```

```
    SUM((b,j), X(a,i,b,j) $ OK(a,i,b,j)) + XX(a,i)
    =E=
       SUM((s),M(s,a,i) $ O(s,a,i));
PODCAP(b,j) $ (SUM(s,D(s,b,j)))..
    SUM((a,i),X(a,i,b,j) $ OK(a,i,b,j)) + Y(b,j)
    =E=
         SUM(s $ D(s,b,j),
            SUM(i $ POE(s,i),
               SUM(a $ REALISTIC(a,i,b,j),
                   M(s,a,i))));
LOAD(s) .. SUM((a,i),M(s,a,i) $ O(s,a,i)) =E= 1;
START .. SUM((a,i) $ SUM(s,O(s,a,i)), XX(a,i)) + Z
                                        =E= SUM((s,i),POE(s,i));
FINISH .. SUM((b,j) $ SUM(s,D(s,b,j)),Y(b,j)) + Z
                                        =E= SUM((s,j),POD(s,j));
ANS .. SUM((a,i,b,j),X(a,i,b,j) $ OK(a,i,b,j)) =E= Z;



MODEL  DFMOD/ALL/;
OPTION solprint=off,sysout=off;
OPTION limrow=0, limcol=0;


$ontext
    These options format the output for a fortran program that
    produces ship schedules.
$offtext
OPTION X:3:0:1;
OPTION XX:3:0:1;
OPTION Y:3:0:1;
OPTION M:3:0:1;
OPTION LINK:3:0:1;
OPTION T:3:0:1;


*SOLVE  DFMOD USING LP MAXIMIZING Z;
SOLVE DFMOD USING MIP MAXIMIZING Z;
$ontext
    These values are used to set the limits of DO Loops within the
    Fortran program, SCHED.
$offtext
SCALARS    XCNT     number of X values
            XXCNT     number of XX values
             YCNT     number of Y values
            MCNT      number of M values
            TCNT      number of possible connections;
XCNT = SUM((a,i,b,j) $ X.L(a,i,b,j),1);
XXCNT = SUM((a,i) $ XX.L(a,i),1);
YCNT = SUM((b,j) $ Y.L(b,j),1);
MCNT = SUM((s,a,i) $ M.L(s,a,i),1);
TCNT = SUM((i,j),1);
DISPLAY XCNT,XXCNT,YCNT,MCNT,TCNT;
DISPLAY XX.L,T,LINK,X.L,Y.L,M.L;
```

APPENDIX B
INPUT FOR EXAMPLE 2

Let there be

- 5 POEs, i = NY, Chr, Jax, Sea, SF
- 5 PODs, j = Eng, Ger, Fr, Kor, Jap
- 30 shiploads
- 60 time periods.

Table 9 and Table 10 are the input data for Example 2.

TABLE 9

$T_{i,j}$ (DAYS)

	Eng	Ger	Fr	Kor	Jap
NY	6	6	6	19	18
Chr	7	7	7	18	17
Jax	8	8	8	18	17
Sea	20	20	20	8	7
SF	19	19	19	9	8

TABLE 10
MOVEMENT REQUIREMENTS

	Eng	Ger	Fr	Kor	Jap
S01.NY.T01*T03	1				
S02.NY.T01*T03		1			
S03.NY.T06*T08	1				
S04.NY.T19*T21		1			
S05.NY.T25*T27			1		
S06.NY.T36*T40	1				
S07.NY.T50*T53			1		
S08.Chr.T02*T04			1		
S09.Chr.T08*T10	1				
S10.Chr.T09*T11			1		
S11.Chr.T24*T26			1		
S12.Chr.T26*T28		1			
S13.Chr.T44*T46		1			
S14.Jax.T02*T04	1				
S15.Jax.T02*T05		1			
S16.Jax.T12*T14	1				
S17.Jax.T23*T25		1			
S18.Jax.T39*T42		1			
S19.Jax.T46*T48			1		
S20.SF.T03*T05				1	
S21.SF.T08*T10				1	
S22.SF.T24*T26					1
S23.SF.T35*T38				1	
S24.SF.T49*T51					1
S25.Sea.T01*T03				1	
S26.Sea.T01*T03					1
S27.Sea.T10*T12	1				
S28.Sea.T12*T14				1	
S29.Sea.T25*T27					1
S30.Sea.T43*T45					1

LIST OF REFERENCES

1. Russell, James A., "Deployment: Will TRANSCOM Make A Difference?", *Military Logistics Forum*, pp. 38-45, June 1987.

2. Interview with Mr. Woody Hall, Technical Director, Systems Analysis Branch, U.S. Transportation Command, MacDill AFB., Florida 33608 and the author, 8 September 1987.

3. Jarvis, John J. and Ratliff, Donald H., *System For Closure Optimization Planning and Evaluation (SCOPE)*, Georgia Institute of Technology, Atlanta, Ga, September 1984.

4. Collier, K. Steven, *Deployment Planning: A Linear Programming Model With Variable Reduction*, M. S. Thesis, Naval Postgraduate School, Monterey, California, September 1987.

5. Dantzig, G. B. and Fulkerson, D. R., "Minimizing The Number Of Tankers To Meet A Fixed Schedule", *Naval Research Logistics Quarterly*, vol. #3, pp. 217-222, September 1954.

6. Bisschop, J., Meeraus, A., "Selected Aspects of a General Algebraic Modeling Language," *Optimization Techniques : Proceedings of the 9th IFIP Conference on Optimization Techniques* Part 2, pp 223 - 233, 1980.

7. Bisschop, J., Meeraus, A., "Toward Successful Modeling Applications in a Strategic Planning Environment," *Large Scale Linear Programming* pp 711 - 745, 1981.

8. Kendrick, D., Meeraus, A., "GAMS : An Introduction," Development Research Department, The World Bank, Washingtion, D.C. 20433, 1987 (revised).

9. Rosenthal, R.E., "Review of GAMS/MINOS Modeling Language and Optimization Program," *OR/MS Today* V. 13, No.3, pp 24 - 32, 1986.

10. Rosenthal, R.E., "Tutorial on the GAMS Modeling Language," Operations Research Department, Naval Postgraduate School, Monterey, CA 93943, 1987.

11. Brooke, Tony and Meeraus, Alexander, *GAMS Documentation*, March 1987.

12. Brown, Gerald G., Graves, Glenn W., and Ronen, David, "Scheduling Ocean Transportation of Crude Oil", *Management Science*, vol. 33, #3, pp. 335-346, March 1987.

INITIAL DISTRIBUTION LIST

		No. Copies
1.	Defense Technical Information Center Cameron Station Alexandria, VA 22304-6145	2
2.	Library, Code 0142 Naval Postgraduate School Monterey, CA 93943-5002	2
3.	Mr. Tony Brooke Analytic Support Unit Development Research Department The World Bank 1818 H. Street, N.W. Washington, D.C. 20433	1
4.	Professor Gerald G. Brown, Code 55Bw Department of Operations Research Naval Postgraduate School Monterey, CA 93943	1
5.	CPT. Michael J. Lally PSC BOX 3500 MacDill AFB, FL 33608-6001	10
6.	Professor Siriphong Lawphongpanich, Code 55 Department of Operations Research Naval Postgraduate School Monterey, CA 93943	1
7.	Mr. Alexander Meeraus Analytic Support Unit Development Research Department The World Bank 1818 H. Street, N.W. Washington, D.C. 20433	1
8.	Commander Mark Mitchell, Code 55Mi Department of Operations Research Naval Postgraduate School Monterey, CA 93943	1
9.	Professor Richard E. Rosenthal, Code 55Rl Department of Operations Research Naval Postgraduate School Monterey, CA 93943	5

10. U.S. Transportation Command 1
TCJD-S-TD
MacDill AFB, FL 33608